

**What is claimed is:**

1. A deterministic method for bootstrapping software onto a remote computing device, comprising:

in response to determining that a connectivity component is missing from the remote computing device, loading the connectivity component via an existing transport mechanism;

using the connectivity component to bootstrap a remote procedure call component, the remote procedure call component for receiving at least one argument via a remote procedure call, and executing a binary stored in a library on the remote computing device; and

loading the library, the library comprising at least one callable binary.

2. The method of claim 1, further comprising determining the connectivity component to load by querying a data store of connectivity components.

3. The method of claim 1, wherein the connectivity component to be loaded onto the remote computing device is determined by a type of central processing unit of the remote computing device.

4. The method of claim 1, wherein the connectivity component to be loaded is determined by a type of platform running on the remote computing device.

5. The method of claim 2, wherein the data store of connectivity components resides on a second computing device.

6. The method of claim 1, wherein the remote procedure call component receives a result of executing the binary stored in the library on the remote computing device to a second computing device.

7. The method of claim 1, wherein the remote computing device is a personal digital assistant.

8. The method of claim 1, wherein the remote computing device is a router.

9. The method of claim 1, wherein the remote computing device is a modem.
10. The method of claim 1, wherein the remote computing device is an OEM board.
11. The method of claim 1, wherein the remote computing device is a smart telephone.
12. The method of claim 1, wherein the binary to be executed in the library is a first version of the binary and a second version of the binary is loaded into the remote computing device library.
13. The method of claim 8, wherein a process running the first version of the binary is terminated.
14. The method of claim 1, wherein the existing transport mechanism is TCP, TCP/IP, KITL®, Emulator or ActiveSync®.
15. A deterministic method for bootstrapping software onto a remote computing device, comprising:

in response to determining that a connectivity component is present on the remote computing device, using the connectivity component to bootstrap a remote procedure call component, the remote procedure call component for receiving at least one argument via a remote procedure call, and executing a binary stored in a library on the remote computing device; and

loading the library using the connectivity component, the library comprising a plurality of callable binaries.

16. The method of claim 15, further comprising determining the connectivity component to load by querying a data store of connectivity components.
17. The method of claim 15, wherein the connectivity component to be loaded onto the remote computing device is determined by a type of central processing unit of the remote computing device.

18. The method of claim 15, wherein the connectivity component to be loaded is determined by a type of platform running on the remote computing device.
19. The method of claim 16, wherein the remote computing device is a first computing device and the data store of connectivity components to be loaded resides on a second computing device.
20. The method of claim 15, further comprising receiving a result of executing the binary.
21. The method of claim 15, wherein the remote computing device is a personal digital assistant, a router, a modem, an OEM board or a smart telephone.
22. The method of claim 15, wherein the binary to be executed in the library is a first version of the binary and a second version of the binary is loaded into the remote computing device library.
23. The method of claim 22, wherein a process running the first version of the binary is terminated.
24. A method for using a connectivity remote procedure call component to enable a computer to execute a function on a remote computing device, the method comprising:
- determining an endpoint of a remote procedure call component on the remote computing device, the remote procedure call component associated with a library, the library comprising a plurality of callable binaries;
  - determining that a remote procedure call component is running on the remote computing device;
  - identifying the function to execute and a library the function is stored in via a remote procedure call to the remote computing device.
25. The method of claim 24, further comprising
- sending the identification of the function to execute and the library the function is stored in at the remote device;

executing the function at the remote device; and

receiving a result of executing the function.

26. A method for using a connectivity remote procedure call component to enable a function on a remote computer to be executed via a remote procedure call from a computer, the method comprising:

receiving an identification of the function to execute and a library the function is stored in at the remote device;

executing the function at the remote device; and

returning a result of executing the function to the computer.

27. The method of claim 26, further comprising

receiving the function to execute and a library the function is stored in via a remote procedure call.

28. A system for implementing distributed software on a remote device comprising:

a connectivity remote call processor for executing a binary on a remote device, determining binaries to be pushed to the remote device and pushing binaries to the remote device; and

a datastore for tracking properties of the remote device.

29. The system of claim 28, wherein the connectivity remote call processor sends information associated with the binary to execute and receives a result of executing the binary.

30. The system of claim 28, further comprising a connectivity server for copying a remote procedure call surrogate onto the remote device.

31. The system of claim 28, wherein the datastore tracks information associated with a CPU of the remote device.

32. The system of claim 28, wherein the data store tracks information associated with a platform of the remote device.
33. The system of claim 28, wherein the data store is indexed by an identifier that identifies the remote device and a surrogate and an endpoint for establishing a connection with the remote device.
34. The system of claim 28, further comprising a bootstrapper for copying at least one binary to the remote device.
35. A system for receiving and executing software on a remote device comprising:
- a connectivity surrogate for receiving information from a computer, executing an indicated binary in a library of binaries and returning a result of executing the function to the computer;
- the library of binaries; and
- a connectivity component adapted to being queried to determine if the connectivity surrogate is loaded and for determining an endpoint of the connectivity surrogate.
36. The system of claim 35 wherein the remote device is a personal digital assistant.
37. The system of claim 35, wherein the remote device is an OEM board.
38. The system of claim 35, wherein the remote device is a smart telephone.